Atty. Dkt. No.: 037145/1001 (NC 37366)

U.S. PATENT APPLICATION

for

MULTIPLEXING OF COMPRESSED CONTROL AND USER-PLANE MESSAGES

Inventor: Pekka Pessi

As Assignor to:

Nokia Corporation
P.O. Box 226
FIN-00045 NOKIA GROUP
Finland

MULTIPLEXING OF COMPRESSED CONTROL AND USER-PLANE MESSAGES

FIELD OF THE INVENTION

[0001] The present invention relates to techniques for end-to-end SigComp compression on a session with intermediate relays.

BACKGROUND OF THE INVENTION

[0002] The Signaling Compression (SigComp) compression protocol (RFC 3320) defines a universal decompression virtual machine (UDVM) and a compression state handler run by a decompressor. SigComp provides a safe environment where the compressing party can ask the decompressor to execute a decompression program and store a decompression state.

[0003] FIG. 1 illustrates two variants of a SigComp message format. A format 12 can include a header section 14, a returned feedback section 16, a partial state identifier section 18, and a remaining SigComp message section 20. The format 12 uses an existing bytecode program. A format 22 can include a header section 24, a returned feedback section 26, a code and destination section 27, a UDVM bytecode section 28, and a remaining SigComp message section 30. The format 22 includes the bytecode within the message.

[0004] The decompression programs are written with the UDVM bytecode. The UDVM bytecode can either be sent within a SigComp message or it can be stored in the decompressor. Bytecode that is stored in the decompressor can be referenced by its identifier, a 6 to 20 byte long hash of bytecode. A compression algorithm can be readily

available in every decompressor, and it can be used by simply specifying a hash code identifying it.

[0005] The SigComp message can be either self-contained or it may refer to states within the decompressor. A self-contained message can be statelessly decompressed. The compressor can get feedback from its UDVM decompressor program. The returned feedback can be piggy-packed as a cookie in the SigComp message sent by the peer compressor.

[0006] FIG. 2 illustrates an example of a SigComp communication system 33. In the communication system 33, a device A is a device B's peer and the device B is the device A's peer. For instance, if the compressor in device A wants to get feedback, it includes the instructions to get it in a bytecode program which it sends to device B. The device B executes the bytecode, and gets a request from device A's bytecode to send a feedback item to the device B. The device B does not need to understand the meaning of feedback item, it is just a cookie. When the device B sends a message to the device A, it attaches the feedback cookie to its own message.

[0007] The decompressor can provide information about its parameters (e.g., supported SigComp version, allowed bytecode execution cycles, available memory size, etc) to the compressor. This parameter information is included in the bytecode. For instance, if device B wants to give its UDVM parameters to the device A, it includes appropriate instructions in the bytecode it will send to device A. The bytecode instructions then generates a well-known returned parameters structure that describes device B's capabilities, and passes it to the device A's UDVM.

[0008] The SigComp decompression algorithm enhances compression by using dictionaries. These dictionaries can be static or dynamic. Static dictionaries are either well known or uploaded by the compressor. Dynamic dictionaries are updated by the decompression algorithm. A decompression algorithm can, for instance, store the message headers to the dictionary, so when a next message is sent, the headers can be referenced efficiently.

[0009] The messaging session relay protocol (MSRP) is a simple means for exchanging Multipurpose Internet Mail Extensions (MIME) messages between two endpoints. Between the endpoints there can be up to two messaging relays. The relays make it possible to establish a messaging session through middleboxes (firewalls, NATs, etc.).

[0010] FIG. 3 illustrates communication between the device A and the device B via relays R_A and R_B. The messaging session protocol in a communication session is transaction-based (request/response). A session carries two kind of request messages: (1) control messages used to establish and authenticate messaging sessions with relays, and (2) user requests messages that actually carry instant messages between endpoints. The relays are typically interested only in control messages. When a messaging session is uncompressed, the relay can determine the type (control or user) of the incoming request. The first few bytes of the message contains the request type and the size of the message. The user messages can simply be forwarded to the outgoing connection without further inspection.

[0011] FIG. 4 illustrates one of the limitations of conventional intermediate elements, or relays, during a messaging session. The messaging stream consists of two kind of messages, control messages that are processed by the relays and end-to-end user

messages that are processed only by the endpoints. If the messaging session is compressed, the relays have to decompress using a decompressor (D) and re-compress using a compressor (C) the messaging stream that flows through them. A relay cannot pass compressed messages through as they have to determine the type of each message. In order to efficiently decompress all messages, they have to maintain separate compression contexts with every endpoint. With a large number of messaging sessions flowing through them, this may require disproportionate amount of resources.

[0012] FIG. 5 illustrates a SIP (Session Initiation Protocol) system where it is possible to compress the message body end-to-end. If hop-by-hop compression is used, the message headers are compressed using hop-by-hop and the compression is applied to the message body twice. It is possible to partially solve the problem by only compressing the low-bandwidth links. Nevertheless, such a solution to the inefficiencies of a SigComp communication system is limited to the scenario where uncompressed messages can be transmitted.

[0013] Thus, there is a need to avoid resource-intensive decompression and compression SigComp systems and methods. Further, there is a need to simplify SigComp states of relays. Even further, there is a need to put multiple user messages in one SigComp message and multiplex control and user-plane messages.

SUMMARY OF THE INVENTION

[0014] The present invention is directed to a method, device, system, and a computer program product where compressed control and user-plane messages are multiplexed. The multiplexing of such messages involves distinguishing between control messages and

user-plane messages using an identifier accompanying the control messages. The identifier signals an intermediate relay that a particular message is a control message and should be decompressed and processed.

[0015] Briefly, one exemplary embodiment relates to a method for communicating messages using a compression protocol. The method includes detecting control messages at a communication intermediary from a compressed stream of messages, decompressing the detected control messages at the communication intermediary, and passing user messages from the compressed stream of messages through the communication intermediary without modifications.

[0016] Another exemplary embodiment relates to a device that communicates messages using a compression protocol. The device includes an input that receives messages, an output that transmits messages, and a processor that detects control messages included in the messages received by the input. The processor decompresses the control messages and directs non-control messages to be communicated through the output without modification.

[0017] Yet another exemplary embodiment relates to a system for communicating messages using a compression protocol. The system includes a first communication device having a compressor and a decompressor, a second communication device having a compressor and a decompressor, and an intermediate relay between the first communication device and the second communication device that detects and decompresses control messages in messages communicated from the first communication device, and passes user messages through to the second communication device without decompression.

[0018] Even another exemplary embodiment relates to a computer program product that has computer code to detect control messages at a communication intermediary from a stream of messages, decompress the detected control messages at the communication intermediary, and communicate user messages from the stream of messages through the communication intermediary without modification.

[0019] Other principle features and advantages of the invention will become apparent to those skilled in the art upon review of the following drawings, the detailed description, and the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] Exemplary embodiments will hereafter be described with reference to the accompanying drawings.

[0021] FIG. 1 is a diagrammatic representation of two variants of SigComp message format.

[0022] FIG. 2 is a diagrammatic representation of communication devices with compressors and decompressors working in a peer relationship.

[0023] FIG. 3 is a diagrammatic representation depicting relay protocol in a communication system where control messages set up connections and control relays and user messages travel end-to-end.

[0024] FIG. 4 is a diagrammatic representation depicting the compression of a relay protocol in which each relay must decompress and re-compress each message it relays.

[0025] FIG. 5 is a diagrammatic representation depicting a communication system where only one communication link with a relay is compressed.

[0026] FIG. 6 is a diagrammatic representation depicting intermediate relays passing end-to-end messages unmodified through them in accordance with an exemplary embodiment.

[0027] FIG. 7 is a diagrammatic representation depicting a message having a byte code pattern at the beginning of a message in accordance with an exemplary embodiment.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

[0028] FIG. 6 illustrates a device 42 communicating via intermediate relays 44 and 46 to a device 48 where the intermediate relays 44 and 46 can pass end-to-end messages unmodified through them. The intermediate relays 44 and 46 detect the hop-by-hop (control) messages from the compressed stream and decompress them. The relays 44 and 46 pass the user messages through without modifications.

[0029] In an exemplary embodiment, the device 42 marks individual compressed messages in the messaging session so that the intermediate relays 44 and 46 on the session can pass through uninteresting (user plane) messages and uncompress only the interesting (control plane) messages. Control messages can be compressed hop-by-hop, user messages end-to-end.

[0030] FIG. 7 illustrates a message 50 according to an exemplary embodiment. The message 50 can include a header section 52, a returned feedback section 54, a code and destination section 55, a MUCCUP bytecode section 56, a compressed algorithm 58 bootstrapped

by MUCCUP, and a remaining SigComp message section 60. Conventional SigComp message formats do not have a ready multiplexing identifier (a port number, for instance). According to an exemplary embodiment, the MUCCUP bytecode section 56 includes a multiplex identifier. MUCCUP refers to Multiplexing of Compressed Control and User-Plane messages. The MUCCUP bytecode is a UDVM program that does not create or access decompression states in the relays. The compressed control messages that relays need to uncompress always carry the MUCCUP bytecode. The bytecode forms a "magic pattern" that can be used to detect control messages.

[0031] Relays 44 and 46 know if its peer supports the MUCCUP bytecode from the first SigComp message sent by the peer. If the peer uses the MUCCUP bytecode in the first message, the relays compress all the control-plane messages using MUCCUP. When a Message Session Relay Protocol (MSRP) relay sees a subsequent SigComp message without the MUCCUP bytecode, it relays the message through unmodified. If the MUCCUP bytecode is there, the relay decompresses the message. If the control messages are used only at the beginning of the session, the relay can go into forwarding mode after initial control messages has been exchanged.

[0032] In alternative embodiments, the MUCCUP bytecode can be used to distinguish hop-by-hop messages and end-to-end messages. End-to-end control messages can be compressed end-to-end, too. The MUCCUP bytecode can be standalone, decompressing the control messages, or it can load the actual compression algorithm and start executing it.

[0033] Referring back to FIG. 1, the format 12 has the partial state identifier section 18. In an alternative embodiment, the

partial state identifier contained in the partial state identifier section 18 can be used as a MUCCUP pattern. Such a partial state identifier can refer to an actual MUCCCUP bytecode. The bytecode can be well-known and locally available in the relays.

[0034] If the SigComp techniques used on a messaging session supports multiplexing of compressed and uncompressed messages, it is possible to send control-plane messages uncompressed and user-plane messages compressed. Further, transition from uncompressed to compressed can be signaled using control messages. All the MSRP messages in a messaging session may be uncompressed, but the user-plane messages may contain SigComp messages in themselves. In other words, the SigComp messages are tunneled through a messaging session.

[0035] There are many alternatives for using the MUCCUP bytecode. For instance, the MUCCUP bytecode can be included in the list of locally available states, the peer may include a specific bit pattern in the returned feedback item, or it may include a special bit pattern in the beginning of the "remaining SigComp message" field. The peer can even include the MUCCUP bytecode in the first message. It is also possible to signal support with external means, for instance, using an SDP attribute.

[0036] This detailed description outlines exemplary embodiments of a method, device, system, and a computer program product for multiplexing compressed control and user-plane messages. In the foregoing description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It is evident, however, to one skilled in the art that the exemplary embodiments may be practiced without these specific details.

In other instances, structures and devices are shown in block diagram form in order to facilitate description of the exemplary embodiments.

[0037] While the exemplary embodiments illustrated in the Figures and described above are presently preferred, it should be understood that these embodiments are offered by way of example only. Other embodiments may include, for example, different techniques for performing the same operations. The invention is not limited to a particular embodiment, but extends to various modifications, combinations, and permutations that nevertheless fall within the scope and spirit of the appended claims.